IN THE CLAIMS:

Claims 1, 6, and 8 are amended herein and new claim 21 is added. All pending

claims and their present status are produced below.

1. (Currently Amended) A method for the direct execution of an XML-document <u>in a data</u>
   <u>processing system</u>, comprising:
   defining the local behavior and process for each element of the XML-document;
   integrating executable instructions with at least one XML-document or a document
type definition (DTD); and
   storing intermediate states of the execution process <u>in a memory of the data</u>
<u>processing system</u> by dynamically creating and redefining a<u>ttributes </u>of element<u>s</u> <u>of the XML</u>
<u>document</u>attributes.

2. (Original)   The method according to claim 1, further comprising:
   (a)   integrating executable instructions by defining for each XML element definition
         and its instances an action made up of executable actions, and actions which are
         references to either the action defined for one of the components of the element
         or to an action defined for any other element of the XML document; and
   (b)   executing an XML-document by executing the action defined for the root of the
         XML document.

3. (Original)   The method according to claim 1, further comprising:
   defining a composition of the action for at least one XML-element definition or
   instance by graphical flow charts.

4. (Original)   The method according to claim 1, further comprising:
   defining the composition of the action for at least one XML-element definition or
   instance in textual form.

5. (Original)   The method according to claim 1, further comprising:
   representing system states in terms of n-dimensional data cubes;

3   providing an open interface by making the n-dimensional cubes readable and

4   writeable for other programming and database systems; and

5   making data structures and functionalities of other programming and database

6   systems accessible using executable instructions.

7

1   6. (Currently amended)   The method according to claim 1, further comprising modules

2   stored in the memory of the data processing system that define a process for each

3   element, where the modules are valid with respect to the following DTD, which is also

4   stored in a memory of the data processing system:

5   <!element module (derived*, expression?, state*, module*>

6   <!attlist module   name CDATA #REQUIRED

7   number CDATA "1">

8   <!element derived (argument*, expression)>

9   <!attlist derived name CDATA>

10   <!element argument EMPTY>

11   <!attlist argument name CDATA>

12   <!element state (action*, transition*)>

13   <!attlist state name CDATA>

14   <!element transition (expression, path)>

15   <!element path (component?)>

16   <!attlist path state CDATA "initial">

17   <!element component (component?)>

18   <!attlist component   name CDATA #REQUIRED

19   number CDATA "1">

20   <!element expression (path | self | src | trg |

21   evalattr | getfirst | getnext |

22   parent | root | apply | external |

23   constant>

24   <!element action (setattr | ifthen | forall | external)>

25   <!element src EMPTY>

26   <!element trg EMPTY>

```
27    <!element self EMPTY>

28    <!element evalattr (expression?)>

29    <!attlist evalattr attribute CDATA #REQUIRED>

30    <!element getfirst (expression?)>

31    <!attlist getfirst attribute CDATA #REQUIRED>

32    <!element getnext (expression?)>

33    <!element parent (expression?)>

34    <!element root EMPTY>

35    <!element apply (expression, expression?)>

36    <!attlist apply op CDATA #REQUIRED>

37    <!element external (expression*)>

38    <!attlist external name CDATA

39    language CDATA >

40    <!element constant EMPTY>

41    <!attlist constant value CDATA #REQUIRED>

42    <!element setAttr (expression?, expression)>

43    <!attlist setAttr attribute CDATA #REQUIRED>

44    <!element ifthenelse (expression, action*)>

45    <!element forall (action*)>

46    <!attlist forall range CDATA "all-elements"

47    variable CDATA>.

48
```

1   7. (Original) A system for use with the method according to one of the preceding claims,

2   comprising:

3   a server providing services to at least one client by executing at least parts of a XML-

4   document according to a XML-robot specification sent from the client to the server or a

5   server providing services to at least one client by sending a XML-robot specification and a

6   XML-document to the client, such that said service is provided by executing of at least part

7   of the sent document on the client according to the sent XML-robot specification.

8

1   8. (Amended) An apparatus for use with the method according to claim 1, comprising:

2　means for receiving from and sending data to a remote computer; means for storing and

3　accessing a XML-document; means for integrating ~~the~~ XML-robot specifications with ~~a~~ the

4　XML-document and means for executing ~~such~~ the integrated document.

5

1　9. (Original) An apparatus for use with the method according to claim 7, comprising means

2　for graphical display of XML-robot specifications within an advanced visual integrated

3　development environment and means for generating XML-documents representing said

4　XML-robot specifications.

5

1　10. (Original) An apparatus according to claim 8 or 9, further comprising means for

2　examining, validating or animating XML-documents or XML-robot specifications.

3

1　11. (Original) An apparatus for the direct execution of XML documents, comprising:

2　　　　means for graphical display of XML-robot specifications within an advanced

3　visual integrated development environment; and

4　　　　means for generating animations of the execution process.

5

1　12. (Original) A method for the direct execution of XML documents comprising:

2　　　　providing an execution specification including

3　　　　　　a DTD;

4　　　　　　graphical flow charts; and

5　　　　　　transition rules;

6　　　　providing an XML document instance including

7　　　　　　an XML document;

8　　　　using the DTD to validate the XML document;

9　　　　constructing an attributed structure tree;

10　　　　decorating the attributed structure tree with the graphical flow charts to create

11　a global flow chart; and

12　　　　executing the global flow chart according to the transition rules to directly

13　execute the XML document.

14

1   13. (Original) A computer-readable medium having computer-readable instructions for

2   performing a method for the direct execution of XML, the method comprising:

3   providing an execution specification including

4   a DTD;

5   graphical flow charts; and

6   transition rules;

7   providing an XML document instance including

8   an XML document;

9   using the DTD to validate the XML document;

10   constructing an attributed structure tree;

11   decorating the attributed structure tree with the graphical flow charts to create

12   a global flow chart; and

13   executing the global flow chart according to the transition rules to directly

14   execute the XML document.

15

1   14. (Original) A computer-readable medium having computer-readable instructions for

2   performing a method for the direct execution of XML-documents, the method comprising:

3   defining the local behavior and process for each element of a XML-document;

4   integrating executable instructions with a document type definition (DTD), an

5   XML-document; and

6   storing intermediate states by dynamically creating and redefining element

7   attributes.

8

1   15. (Original) A system for the execution of an XML document comprising

2   an interpreter generator having an input and an output, the input operative to

3   receive an XML specification, the interpreter generator operative to produce at the output an

4   interpreter, the interpreter having an input and an output, the input operative to receive an

5   XML document, the interpreter operative to validate the XML document with respect to a

6   general DTD and to execute the XML document .

7

1   16. (Original) A system for the execution of an XML document comprising:

2 a compiler generator having an input and an output, the input operative to

3 receive an XML specification, the compiler generator operative to produce at the output a

4 compiler, the compiler having an input and an output, the input operative to receive a XML

5 document valid with respect to a general DTD, the compiler operative to produce an

6 executable document at the output.

7

1 17. (Original) A system for the execution of an XML document comprising:

2 a first interpreter having an input, the input operative to receive a XML

3 specification:

4 a second interpreter coupled to the first interpreter, the second interpreter

5 having an input, the input operative to receive a XML document valid with respect to the

6 general DTD, the first interpreter starting a process in the second interpreter, the second

7 interpreter operative to execute the XML document.

8

1 18. (Original) A system for the execution of an XML document comprising:

2 an interpreter having an input, the input operative to receive a XML

3 specification, the interpreter operative to interpret the XML specification;

4 a compiler coupled to the interpreter, the compiler having an input and an

5 output, the input operative to receive an XML document, the interpreter operative to start the

6 compiler; the compiler operative to generate an executable XML document on the output.

7

1 19. (Original) A method for the execution of an XML document comprising

2 (a) setting a global variable cur to a root reference;

3 (b) setting the value of a global variable mod to refer to a module element

4 describing the execution behavior of the root;

5 (c) copying all state and derived elements from the module mod into the

6 element cur, setting the attribute origin of all state and derived elements to cur;

7 (d) copying the state and derived elements of the sub-modules of module mod

8 into the corresponding components of element cur;

9 (e) update cur to cur.traverse; and

10 (f) if cur is undefined then executing the XML document else returning to (a).

11

1   20. (Original) The method according to claim 19, wherein executing the XML document

2   comprises:

3   (i) setting cur to the XML document's root;

4   (ii) setting a global variable curstate to initial;

5   (iii) iterating a variable state over all state elements of cur;

6   (iv) if a name attribute of state matches curstate then setting cur to the value of

7   attribute origin of state else terminate execution;

8   (v) iterating over all actions inside state;

9   (vi) resetting cur to its original value; and

10   (vii) returning to (iii).

11

1   21. (New) A method for the direct execution of an XML-document in a data processing

2   system, comprising:

3   defining the local behavior and process for each element of the XML-document;

4   integrating executable instructions with at least one XML-document or a document

5   type definition (DTD); and

6   storing intermediate states of the execution process in a memory of the data

7   processing system by dynamically creating and redefining elements.

8

9